Application of Job-shop Scheduling Based on Hybrid Genetic Algorithm

Guang Dong[1], Wei He[2], Ming Fang[3], Guangcai Cui[4]

School of Computer Science and Technology, Changchun University of Science and Technology, China

[1]lightdg@cust.edu.cn; [2]hw@cust.edu.cn;

[3]fangming@cust.edu.cn; [4]gccui@cust.edu.cn

***Abstract*-This paper designs a hybrid genetic algorithm for job-shop scheduling combined with a Genetic Algorithm (GA), Simulated Annealing (SA), and a Tabu Search (TS) algorithm. It gives a method for initial population generation for avoiding unreasonable solutions, and introduces SA and TS into the genetic operation mechanism of GA for overcoming the lack of job-shop scheduling optimization. Simulated computations show the superiority of the hybrid genetic algorithm, and examples verify the algorithm's efficiency.**

Keywords-Job-shop Scheduling; Genetic Algorithm; Simulated Annealing Algorithm; Tabu Search

I. INTRODUCTION

Because any job-shop scheduling algorithm has different degrees of advantages and disadvantages, researchers in recent years have combined various algori-thms to utilize advantages and to avoid disadvantages for achieving optimal scheduling. Typical algorithms include the choice of combining greedy and simulated annealing algorithms; an algorithm of this type searches by using a greedy algorithm, makes the obtained job as initial JIE, and then uses the simulated annealing algorithm to resolve a machine scheduling problem. Experimental results show that this method is better than using just simulated annealing or greedy algorithms [1]. Other options combine simulated annealing with heuristic algorithms to resolve job-scheduling problem with delivery [2], and combine genetic with tabu algorithms to solve earliness/tardiness schedu-ling problem with waiting casts [3], Melt-Annealing algorithm, an improved simulated annealing method for flow shop problems proposed in reference [4] analyses asym-ptotic convergences to global optimal solution by six different random sampling patterns, and solves the sort of problem of Flow Shop with a minimal Makespan index, Unlimited Intermediate Storage (UIS), finite Intermediate Storage (LIS), and No Intermediate Storage (NIS).It is an improved GA method proposed in reference [5] by minimizing the maximum completion time, which introduces machine selection items in chromosome coding to improve equipment utilizations and balances, or the chromosome encoding method used in reference [6], which generates different codes according to various processes.

Let numbers represent the process of one work piece. All these distinct numbers are coded for evolutionary computation, and a matrix decoding method is proposed using a multi-objective job-shop scheduling algorithm as in reference [7], which combines positive-reverse scheduling with a two-population genetic algorithm.

To improve the local search ability of GA, also to improve further optimization quality and search efficiency, and compensate for the lack of a single optimization method, this paper gives a GA-SA-TS algorithm, which combines GA, and tabu search with simulated annealing algorithms. Examples are given to verify the efficiency of the algorithm.

II. MATHEMATICAL MODE OF JOB-SHOP SCHEDULING

Job-shop scheduling is a simplified model of many realistic production scheduling problems, and it is a typical NP-hard problem. Job shop is a production procedure using workshop resources for a certain object, while the job scheduling effectively sorts the job shop and minimizes (or maximizes) a certain objective function. Here, a job object can be a component, or a part of the component; it can be one part of a process or a certain process step. Because the process is a basic element of components and parts, the work of the same process step is usually an absolute division of process. That is, these will be finished on the same machine. Therefore, we established the job shop object as a part of the process. Certainly, job scheduling objects are established in the process as well.

A. Basic Constraints of Job-shop Scheduling

Job-shop scheduling not only has the extreme property, that is, its result is the minimum (or maximum) value of a solution, it also has the effectiveness property, that is, its result must meet the basic constraints of job-shop scheduling shown as follows:

- Every process must be processed and produced on its specified machine.
- Every process must wait for entering into a process procedure until its previous process has been finished.
- A machine only can process one part at a time.
- The procedure of processing parts on the machine is not allowed to be interrupted.

B. Mathematical Model

To describe the mathematical model of JSSP, we first give the following hypothesis and definition.

Assume that P denotes a collection of n parts; then P can be described as P= {P1, P2, P3⋯, Pn}.

Assume that M denotes a collection of m machines; then M can be described as M= {M1, M2, ⋯, Mm}.

Assume that the process number of Pi is Ki, and JPi denotes the process collection of Pi part; then JPi can be denoted as JPi= {Ji (1), Ji (2),⋯, Ji (k)}.

Definition 1: Suppose the elements of process Ji belonging to Pi on M as: Mi(1), Mi(2)…, Mi(k), denoted by using i(1), i(2)…, i(k), where, k=max{k1, k2…,.kn}, $0 \leq (i(j)) \leq m$,

$0 \le i \le n$, $0 \le j \le k$. Then, the matrix of a machine process order can be described as:

$$Q=\begin{bmatrix} 1(1) & 1(2) & \cdots & 1(k) \\ 2(1) & 2(2) & \cdots & 2(k) \\ \cdots & \cdots & \cdots & \cdots \\ n(1) & n(2) & \cdots & n(k) \end{bmatrix}$$

Definition 2: Suppose that the process order on machine Mi is JMi, where 0<i<m; then JMi can be denoted as JMi= {j(1),j(2)…, j (li)}. Here, li is the total number of process orders on Mi, and j(i) is the order number of the i^{th} processed part.

Definition 3: Matrix JM denotes all the arrangement of machine processes; we call JM to be the arrangement matrix of the machine process. Formally, it is:

$$JM=\begin{bmatrix} 1(1) & 1(2) & \cdots & 1(w) \\ 2(1) & 2(2) & \cdots & 2(w) \\ \cdots & \cdots & \cdots & \cdots \\ m(1) & m(2) & \cdots & m(w) \end{bmatrix} \text{ or } JM=\begin{bmatrix} JM_1 \\ JM_2 \\ \cdots \\ JMm \end{bmatrix}$$

Here, JMi denotes the process order on the i^{th} machine.

Definition 4: We define the process time matrix T as follows:

$$T=\begin{bmatrix} t1(1) & t1(2) & \cdots & t1(w) \\ t2(1) & t2(2) & \cdots & t2(w) \\ \cdots & \cdots & \cdots & \cdots \\ tm(1) & tm(2) & \cdots & tm(w) \end{bmatrix}$$

Each element in the matrix corresponds to an element in JM. Let (tj (w)) denote the finish time of j (w).

Assume that P is the set of parts, M is the machine set, Q is the matrix of machine process order, and JM is the order matrix for the machine process.

For a specified machine processes matrix JM0, if it meets the formula T(JM0) = minT(JM), that is, JM0 minimizes the objective function F(JM), and compatibilities with Q, then we call JM0 as JSSP optimal solution; call the procedure of solving JM0 as job-shop scheduling. Its objective function is:

$$F = \min(T(J_m)) = \min \max_{1\le i\le n}\left\{ \max_{1\le j\le v}\{T_{ij}\} \right\}$$

Here, w=max $\{k_i\}$, k_i is the order number of workpieces J_i, T_{ij} denotes the required process time of the j^{th} process of the i^{th} workpiece.

III. HYBRID GENETIC ALGORITHM

As a kind of group optimization, a genetic algorithm optimizes the solution set through mutation or restructuring operations, and implements fast search operations ultimately. But, it has early-maturing and poor local search ability problems. Simulated annealing can be introduced into a cross-selection mechanism of a genetic algorithm to avoid the shortcomings of early-maturing, when high temperature (pre-GA) similar to the individual fitness of offspring produces similar crossover probability, and when temperature falls, the enhanced role of stretches, similar to the fitness of individual fitness differences, amplification allowing excellent individual advantage is more obvious [8]. Variation of pressure with TS to solve the problem GA, GA mutation probability is difficult to solve the contradiction selected, and because the tabu search algorithm to avoid the circuitous search capability, enhanced local search hybrid algorithm to improve the convergence speed.

A. GA-SA-TS Hybrid Strategy

GA-SA-TS hybrid strategy is a double-layer parallel search structure. At process level, the hybrid algorithm carries out GA, SA and TS search under various temperatures. It is a kind of double-layer serial structure. Here, the initial solution results of SA come from the cross evolution results of GA; the solution of SA obtained by Metropolis sampling process will become further evolution of the population of GA. The initial solutions of TS come from the evolution of variation of GA, the solution obtained by TS also become a future evolution of population of GA. At space laver, GA provides a parallel search structures, it transforms TS, SA into parallel TS, SA algorithms. Therefore, the hybrid algorithm introduced in this paper is optimizing populations from parallel structure.

GA-SA-TS hybrid strategy of search behavior is controlled. It can be controlled by initial temperature of SA, anneal function, sampling frequency and the length of TS tabu table. Control of initial temperature, initial search behavior can control algorithm, controlling the temperature level can control the strength of the jumping ability of an algorithm. The jumping ability under high temperature is helpful for avoiding premature convergence. High temperature high jump is beneficial to avoid falling into local minima; chemotaxis optimization under low temperature is helpful to improve local search ability. Controlling the rate of temperature decreasing can control the decreasing amplitude of jumping ability and influence the smooth of search process. Controlling sampling frequency can control the search ability under various temperature levels, and influence the corresponding cross probability distribution.

Controlling and setting the length of tabu table can improve the performance of the algorithm. In short, controllable performances enhance the low ability of the local search of GA, and overcome the premature shortcomings.

B. Key Technologies and Main Steps of Hybrid Genetic Algorithm

Using hybrid genetic algorithm to solve the key technologies of job-shop scheduling includes the designs of genetic operator and tabu search.

1) Process Based Encoding Method:

The encoding method need to be carefully designed when using genetic algorithm to resolve the job-shop scheduling. The adopted encoding method must ensure that the scheduling results are feasible scheduling solution. This paper selected a process based expression method after considering various factors involved in encoding procedure. This kind of expression method encodes scheduleing into a sequence of the process. It makes each gene to represent a process, each gene chain to represent a possibility of sorting a group tasks.

Assume that there are n processes in a group of tasks; it is not specified for each process from 1 to n in encoding process (that is, using different codes represents different process). Because of the limitation of processing technology, this process method results that chromosome produces infeasible scheduling solution, and increases difficulty in operation.

Base on the reasons above, in this paper, we assign the same codes for the process belonging to the same part, and explain them according to the appearance order in given chromosome. For n workpieces and m machines problems, there are n=m genes in a chromosome; each workpiece appears m times in chromosome, each gene does not indicate a specific process of the workpiece, but indicate a process with upper and lower dependency. It is easy to observe that the arbitrary arrangement of a chromosome always produce a feasible scheduling. In the following, a practical example is given for explaining the coding ideas.

Workpiece 1: (1, 1, 1, 2) (1, 2, 2, 3) (1, 3, 4, 4) (1, 4, 1, 6)

Workpiece 2: (2, 1, 5, 7) (2, 2, 4, 8) (2, 3, 2, 3) (2, 4, 3, 9) (2, 5, 1, 4)

Workpiece 3: (3, 1, 2, 6) (3, 2, 4, 7) (3, 3, 1, 1) (3, 4, 5, 2) (3, 5, 4, 7) (3, 6, 1, 9)

Workpiece 4: (4, 1, 3, 1) (4, 2, 2, 10) (4, 3, 1, 2)

Workpiece 5: (5, 1, 2, 13) (5, 2, 3, 15) (5, 3, 4, 1) (5, 4, 3, 1)

Workpiece 6: (6, 1, 2, 3) (6, 2, 4, 3) (6, 3, 5, 6)

Here, the symbols in brackets above denote the number of workpiece, the order of process of the workpiece, the machine tool number used by the process and the process time of the process respectively.

A chromosome consists of a group of digits, using a digit denotes a process of the workpiece. However, the same digit which appears in different position in chromosome denotes a different process of the workpiece. So, the chromosome according to this example consists of a encoded string which includes four 1, five 2, six 3, three 4, four 5 and three 6, etc. such as 221321251436123556654343, where the digits i appearing at j times denotes that the j^{th} process of workpiece i. Therefore, all the process of ij can be denoted by using symbol i. As shown in the following table, we explain the chromosome in the example as a feasible scheduling. The chromosome in the instance can be explained as feasible scheduling in Table I:

TABLE I
CHROMOSOME EXPLANATION

Gene	2	2	1	3	2
Process	(2,1,5,7)	(2,2,4,8)	(1,1,1,2)	(3,1,2,6)	(1,3,4,4)
Gene	1	2	5	1	4
Process	(2,2,4,8)	(1,4,1,6)	(5,1,2,13)	(1,3,4,4)	(4,1,3,1)
Gene	3	6	1	2	3
Process	(3,2,4,7)	(6,1,2,3)	(1,4,1,6)	(2,5,1,4)	(3,3,1,1)
Gene	5	5	6	6	5
Process	(5,2,3,15)	(5,3,4,1)	(6,2,4,3)	(6,3,5,6)	(5,4,3,1)
Gene	4	3	4	3	3
Process	(4,2,2,10)	(3,4,5,2)	(4,3,1,2)	(3,5,4,7)	(3,6,1,9)

2) Cross Operators as the Most Important Operators:

For the chromosome individual based on ordering processes, it produces non-means encoding string if adopting traditional single-point crossover operators. That is, it produces infeasible scheduling solution. As the following, this paper adopts the crossover operators which combine single-point with order crossovers, and describe the procedure combining with practical examples given in the encoding procedure.

Step1. In the two parent chromosomes popchrom1 and popchrome2 (hereinafter, P1 and P2), we randomly select an integer smaller than Number (total number of genes belonging to a chromo-some), from the first gene to i^{th} gene are considered as crossover region, and the encoding string in this region is denoted with T1 and T2. Now, there are two parent chromosomes:

P1: 51253231613336631442225 45

P2: 255321161252444532333 63361

Randomly select i=6, then the coding string of the crossover region is:

T1: 5 1 2 5 3 2

T2: 2 5 5 3 2 1

Step2. In P1, search all the gene values in T2 from left to right, set the gene value corresponding to the position (digits of italics add thickly) of P1 to 0. Similarly, in P2, search all the gene values in Tl from left to right, set the gene value to 0 as well.

Pl: 510000306133366314422 2045

P2: 205320161002444500336 3361

Step3. To concentrate the gene with zero value of p1 and p2 into the crossover region by left shifting, while keep the other genes with a non-zero value the same relative order.

Pl: 000000513613336631442 2245

P2: 000000253216124445336 3361

Step4. To replace the content of P1 crossover region with the content of T2, replace the content of P2 with the content of T1. So far, two child individual child1 and child2 can be obtained respectively.

3) The Main Steps are Described as the Following when Applying Hybrid Genetic Algorithm to Solve Job-Shop Scheduling:

Assume S is a solution set consisting of all possible solutions. For arbitrary $i \in S$, it has a determined objective function value f (i). Schedule is used to save current optimal chromosome. TotalTime is an optimal individual fitness value.

Here, the iterative target is solving minimum value of the function based on the feature of sche-duling problem. Its main steps are:

Step1. Set the size of initial population as pop-size, scale of crossover and variation with Pc and Pm respectively; set cooling schedule para-meters of SA under the concept of quasi-equilibrium: initial temperature T0, temperature attenuation coefficient α, set the number of can-didate solutions and tabu table length of TS algorithm dynamically,

set the adaptive optimal scheduling best and coincided FitValue.

Step2. Using for process based decimal encoding products popSize initial feasible solutions, let the initial search algebra *gen = 0*.

Step3. Application of heuristic rules on the chromosome decoding, arranging the process order of each part to be processed on each machine, computing and evaluating the fitness function value of the current population.

Step4. Determine whether a termination condition is satisfied. If the termination condition is satisfied then the algorithm ends else continue.

Step5. Let $T_{k+1}=\alpha T_k$, $gen^{+}=1$, according to crossover probability Pc, using simulated annealing crossover operators products new individuals, to decode and compute fitness value for it.

Step6. On the solution to the mutation probability Pm for screening, for each individual tabu mutation screening:

6-1. from the current population selected solu-tion into the tabu search stack.

6-2. pop a solution *x[i]* from tabu search stack, set tabu table as null, optimal state as null.

6-3. dynamic generation neighbourhood solution and the number of candidate solutions from the current solution.

Check whether candidate solutions meet contempt criteria or not. If meet, then replace the optimal state *y* which meets the contempt criteria with *x[i]*, forms current new solutions, that is, *x[i]=y*, put *y* corresponding tabu target into tabu table, and release the targets which meet tabu length *len* from tabu table, replace optimal state with *y*, then return to step6, otherwise return to step5.

6-4. check candidate solution corresponding to tabu features; select the non-tabu target in candidate solution set corresponding optimal state as new current solutions. Meanwhile, put the correspondding targets into tabu table, and release those targets meeting tabu length from tabu table.

6-5. if the required number of tabu search has been met, then ends the searching for *x[i]*, jump to step7, otherwise return to step3.

6-6. if all the solutions in the tabu search set have been carried out in tabu search, then stop carrying out the tabu search algorithm, otherwise return to step2.

Step7. Save the current solution with best performance into Best, fitness value into FitValue.

Step8. Return to Step4.

Explanation1. The number (*candi*) of candidate solutions in this paper is related to the scale of the problem, it is generated dynamically. The following is the rules:

$$candi = \begin{cases} n & 0 < n \le 25 \\ 2n & 25 < n \le 50 \\ 10n & 50 < n \le 100 \end{cases}$$

Explanation2. Dynamic generation tabu table length *len* as the following rules:

$$len = \begin{cases} 4 & 0 < n \le 20 \\ 5 & 20 < n \le 60 \\ 6 & 60 < n < 100 \\ 7 & n = 100 \end{cases}$$

C. Experiments

For small scale production like data MT10, the sorted GANTT figure by device sequences can be obtained after system running, as shown in Fig 1.

To investigate the performance of hybrid strategy of GA-SA-TS, and compare GA, SA, TS, we simulated 11 typical examples for 30 times through example simulation module. Table II summaries the obtained optimal solutions and 30 times simulated average values.

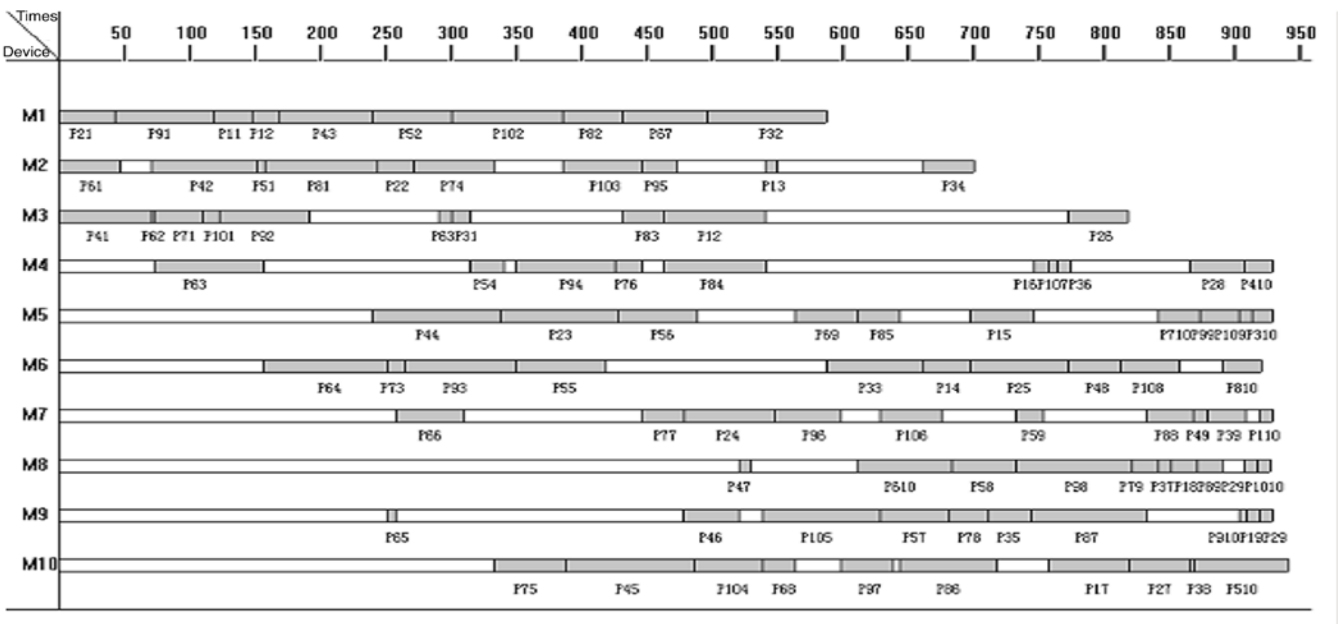

Fig. 1 Sorted GANTT figure by device sequences

TABLE II
PERFORMANCE COMPARISON OF GA-SA-TS, GA, SA AND TS

Problem	C**	GA-SA-TS		GA		SA		TS	
		C*	%	C*	%	C*	%	C*	%
MT06	55	55	0	55	0	55	0	55	3.273
LA01	666	666	0	666	0	666	0	666	3.048
LA06	926	926	0	926	0	926	0	926	0
LA11	1222	1222	0	1222	0	1222	0	1222	0
MT10	930	930	2.34	997	11.87	939	6.65	962	10.89
MT20	1165	1165	1.12	1247	12.40	1227	8.33	1238	11.41
LA16	945	945	0.5	979	6.032	979	4.17	979	5.032
LA21	1046	1046	1.147	1156	14.56	1083	7.49	1157	15.53
LA26	1218	1220	1.181	1328	12.78	1253	5.10	1290	10.78
LA31	1784	1790	0.785	1836	4.826	1784	0.39	1785	1.234
LA36	1268	1268	0	1384	11.91	1321	5.80	1302	4.236

The 11 typical examples include: MT06(6×6), MT10(10×10), MT20(20×5) and LA01(10×5), LA06(15×5), LA11(20×5), LA16(10×10), LA21(15×10), LA26(20×10), LA31(30×10), LA36(15×15), where MT and LA represent question constructors Mute, Thompson and Lawrence, and the digits in brackets represent the number of parts and the number of machines respectively[9].

On the optimization of GA-SA-TS index and GA, SA and TS for comparison, the statistical results are shown in Table II. In the Table, the meaning of each symbol, C * * is for the optimal solution of the problem. The C * algorithm simulation of the optimal value is of 30% average optimal value relative to C * * deviation.

IV. CONCLUSIONS

In this paper, the researched hybrid genetic algorithm selected practical problems as target; it has a good effect on solving multi-workpiece, multi-process and multi-machine mixing flow processes; it also has a better application value for improving the efficiency of job-shop scheduling.

ACKNOWLEDGMENT

Herewith, we thank all our colleagues who gave us support and advice, especially Professor Hua-min Yang from the School of Computer Science and Technology, Changchun University of Science and Technology of China.

REFERENCES

[1] G. Foo, "Future paths for integer programming and links to artifical intelligence", *Computer & Oper.Res*, vol. 13, pp. 533–548, Mar. 2006.

[2] M. Languna and W. Barnes, "Intelligent scheduling with tabu search: An application to jobs with linear delay penalties and sequence-dependent setup costs and times", *Journal of Applied Intelligence*, vol. 3, pp. 159–180, 2004.

[3] J. Wang, K. Ma and T. Feng, "The application of adaptive immune genetics algorithmson hybrid-process job-shop schedule", *J. of Xian University Engineering Science and Technology*, vol. 19, pp. 79–81, Jan. 2005.

[4] L. Wang and D. Z. Zheng, "Study on unified framework of hybrid optimization strategies", *Control and Decision*, vol. 17, pp. 33–36, Jan. 2007.

[5] W. L. Wang and Q. D. Wu, "Modified adaptive genetic algorithms for solving job-shop scheduling problems", *Manufacturing Technology & Machine Tool*, vol. 10, pp. 104–108, Oct. 2010.

[6] H. R. Zhou and P. E. Zheng, "Method for GA-based solution to job shop scheduling optimization", *Application Research of Computers*, vol. 25, pp. 2291–2294, May. 2008.

[7] W. L. Wang and J. F. Li, "Double-population genetic algorithm for multi-objective Job Shop scheduling problem", *Computer Integrated Manufacturing Systems*, vol. 17, pp. 808–815, Sep. 2011.

[8] Y. Cho and S. Sahni, "Preemptive scheduling of independent jobs with release and due times on open", *Flow and Job Shops, Oper. Res.*, vol. 29, pp. 511–522, Nov. 2006.

[9] online ftp://mscmga.ms.ic.ac.uk/pub/jobshop1.txt.

Guang Dong In December 1963, he was born in Jilin of China, in July 1985, graduated from Dalian University of Science and Technology of china, the department of computer science and engineering, computer science professionals, and obtains the technology bachelor's degree.
He is currently an associate professor, Master tutor, and worked at school of computer science and technology in Changchun University of Science and Technology, Changchun, China. He mainly engages in computer science and technology, embedded system and database technology application of teaching and research work.
Prof. Dong is the member of Chinese Computer Academic society.

Wei. He In November 1978, she was born in Jilin of China, in July 2000 graduated from Changchun University of Science and Technology, the department of computer science and technology, the computer science and technology professionals, and obtains the technology bachelor's degree, in June 2003, graduated from Changchun University of Science and Technology, the department of computer science and technology, the computer science and technology professionals, and obtains the technology Master's degree.
She is currently a lecturer, and worked at school of computer science and technology in Changchun University of Science and Technology, China. She mainly engages in computer science & technology and medical image processing of teaching and research work.

Ming. Fang In February 1977, he was born in the Jilin of China, in July 2000 graduated from Changchun University of Science and Technology, the department of computer science and technology, the computer science and technology professionals, and obtains the technology bachelor's degree. In 2011, Hokkaido University, obtains doctor's degree in the technology
He is currently a lecturer, and worked at school of computer science and technology in Changchun University of Science and Technology, China. He mainly engages in computer science & technology and Computer vision of teaching and research work.

Guangcai. Cui In December 1964, he was born in Heilongjiang of China, in July 1986, graduated from Beijing University of Science and Technology of china, the department of computer, computer science professionals, and obtains the technology bachelor's degree. In 2004, Jilin University, obtains doctor's degree in computer
He is currently a professor, Doctoral tutor, and worked at school of computer science and technology Changchun in University of Science and Technology, Changchun, China. He mainly engages in the artificial intelligence and Chinese information processing of teaching and research work.
Prof. Cui is the member of Chinese Computer Academic society, Jilin province, managing director of the computer society.